

Solución de un Problema de la Tardanza Total Ponderada en el Modelo Básico, Usando Un Algoritmo De Búsqueda Tabú

Elkin RODRÍGUEZ
Diego CARMONA
Luis Fernando MORENO
Francisco Javier DÍAZ
Gloria Elena PEÑA
Carolina CÁRDENAS

Escuela de Sistemas, Facultad de Minas, Universidad Nacional de Colombia

Medellín, Antioquia, Colombia

RESUMEN

Este artículo presenta un algoritmo de búsqueda tabú para resolver el problema de la Tardanza Total Ponderada en el Modelo Básico (SMTWTP). El algoritmo desarrollado emplea un vecindario basado en intercambios aleatorios pareados. Para realizar los experimentos de prueba, se fijaron diferentes tamaños de vecindario. Los resultados muestran que el algoritmo produce soluciones óptimas o cercanas al óptimo en los problemas de prueba analizados. Los distintos tamaños del vecindario no muestran diferencias significativas en la frecuencia de hallazgos de soluciones óptimas. Se encontró que el valor del criterio de paro *número de iteraciones sin mejorar*, se puede reducir en cerca de un 50%.

Palabras clave: tardanza total ponderada, búsqueda tabú, intercambios aleatorios pareados, modelo básico de programación, secuenciación.

1. INTRODUCCIÓN

La técnica de búsqueda tabú es considerada en la actualidad, gracias a los resultados reportados por muchos autores, como una de las mejores heurísticas para solucionar problemas de optimización combinatorial. Este trabajo presenta un procedimiento de búsqueda tabú para el problema de minimización de la tardanza total ponderada en el modelo básico.

Este artículo se encuentra estructurado como sigue: en la sección 2 se presenta el estado del arte, donde se presenta una breve revisión de las investigaciones realizadas hasta la fecha; en la sección 3 se realiza el planteamiento del problema de la tardanza total ponderada para el modelo básico (SMTWTP). En el apartado 4 se presenta la descripción de la técnica de búsqueda tabú utilizada para resolver el problema, en la sección 5 se definen los parámetros de entrada del enfoque propuesto. Los resultados de la investigación son reportados en la sección 6. Las secciones 7 y 8 presentan las conclusiones y el trabajo futuro respectivamente.

2. ESTADO DEL ARTE

El SMTWTP es NP hard y problemas con más de 50 trabajos no pueden ser resueltos de manera óptima con los algoritmos de ramal y límite. Por lo tanto, varios métodos heurísticos han sido propuestos para su solución. Estos incluyen heurísticas de construcción simple como EDD o SPT y metaheurísticas como optimización a través de colonias de hormigas, enfriamiento simulado, búsqueda tabú, algoritmos genéticos y búsqueda local iterada ILS.

Estos métodos metaheurísticos han tenido gran aplicación en los diferentes problemas de programación como job shop, flow shop, flexible job shop, etc. A continuación se presenta una breve reseña de los trabajos que se han desarrollado en el pasado para SMTWTP.

El método de búsqueda local tiene varias técnicas entre las que se encuentran ILS, Random Swap, Sliding Mutation, Variable Neighborhood y Neighborhood Interchange. Algunas de éstas han sido trabajadas para el problema del SMTWTP por [3], [6] y [12].

Storter, Selcuk y Akturk [14] desarrollaron un algoritmo genético conocido como PSGA (Problem Space Genetic Algorithm) para resolver el problema del SMTWTP. Esta técnica se ha trabajado para problemas de programación de Job Shop con buenos resultados.

La Colonia de Hormigas es una metaheurística de desarrollo reciente y se han construido aplicaciones con resultados prometedores para el problema del agente viajero, el problema de ordenamiento secuencial, enrutamiento de vehículos y problemas de enrutamiento en ambientes altamente dinámicos. Su aplicación al problema del SMTWTP ha sido llevada a cabo por [4], combinando el uso de colonias de hormigas y la búsqueda local. Existe amplia literatura que muestra la aplicación de búsqueda tabú en problemas de programación de tareas. Respecto al modelo básico (single machine), [5] comparan el algoritmo de búsqueda tabú con tres heurísticas de construcción; [8] proponen un enfoque de búsqueda tabú para minimizar la tardanza media y [9] presentan métodos de búsqueda tabú para el modelo básico con tiempos de preparación secuencialmente dependientes, en problemas hasta de 60 trabajos y vecindarios basados en dos tipos de movimientos.

3. DESCRIPCIÓN DEL PROBLEMA

El problema de la tardanza ponderada para el modelo básico se puede plantear así: un conjunto de n trabajos disponibles en el tiempo cero, deben ser secuenciados sin interrupción en una única máquina que sólo puede procesar un trabajo a la vez. Cada trabajo j tiene un tiempo de procesamiento entero p_j , una fecha prevista (o de compromiso, o entrega) d_j y una penalización w_j por cada unidad de tiempo que el trabajo finalice después de la fecha de entrega $d_j > 0$.

Una vez secuenciados todos los trabajos, cada uno tiene un tiempo de terminación C_j . Se define T_j como la tardanza del trabajo. Formalmente, ésta puede ser definida como:

$$T_j = \text{Max}(0, C_j - d_j) \quad (1)$$

Donde $E_c(1)$ representa la diferencia positiva a partir de la fecha de entrega. Si se asume que la función de penalización es lineal, formalmente, el problema consiste en hallar una secuencia s que

minimice:

$$f(s) = \sum_{j=1 \text{ to } n} w_j T_j \quad (2)$$

Pinedo [11] demuestra que el problema es NP-completo. Diversos enfoques han sido planteados para resolver el problema, incluyendo métodos matemáticos de programación dinámica y heurísticos sencillos y sofisticados.

4. TÉCNICA UTILIZADA

Con el fin de dar una solución eficaz al problema de la tardanza ponderada para el modelo básico, se ha adoptado la técnica heurística de búsqueda tabú, pues ofrece grandes ventajas respecto a otras heurísticas estudiadas y ha demostrado, tanto en este estudio como en los adelantados por otros autores (ver apartado de estado del arte), ser una técnica prometedora que arroja muy buenos resultados, cuyo potencial de alcanzar el óptimo en varios problemas de la literatura es bastante bueno.

La búsqueda tabú se basa en el uso de estructuras de memoria para escapar de los óptimos locales, donde se almacenan las últimas soluciones exploradas o los movimientos realizados (en el caso de trabajar sólo con *atributos* representativos del movimiento), y que puede ser utilizada con el fin de impedir la evolución hacia soluciones ya visitadas al moverse por el espacio de búsqueda [7].

La estructura de memoria del algoritmo está constituida por la denominada *lista tabú*. Cada vez que una solución es encontrada o en su defecto, un movimiento es realizado, se introduce en una lista circular (estrategia FIFO, la primera en entrar es la primera en salir) de un tamaño determinado, de manera que los elementos de dicha lista se encuentran penalizados o en estado "tabú" durante un cierto tiempo. Por lo tanto, una solución no será aceptada siempre y cuando permanezca en la lista tabú, pero de no estar, siempre se acepta a pesar de que la nueva solución sea peor que la anterior.

La memoria empleada por el algoritmo puede ser de corto o de largo plazo. La memoria de corto plazo, cuyo principal objetivo es explorar profundamente una región determinada del espacio de soluciones, almacena secuencias (o atributos de éstas) que fueron visitadas recientemente; mientras que la memoria de largo plazo almacena la frecuencia con que ocurren los atributos en las soluciones visitadas, esto con el fin de identificar regiones.

Si todos los movimientos de la lista tabú se prohíben se evita entrar en ciclos, pero se pueden perder movimientos que acerquen a mejores soluciones. Por ello se ha introducido el concepto de *nivel de aspiración*, que supone un criterio para aceptar soluciones que están incluidas en la lista tabú, pues en ocasiones parece no tener sentido rechazar un movimiento tabú que diese lugar a una solución cuyo valor fuese menor que el mencionado nivel de aspiración.

En los algoritmos basados en búsqueda tabú existen dos estrategias importantes, estas son las estrategias de diversificación y de intensificación. Las estrategias de intensificación se basan en la escogencia de reglas que permitan concentrarse y examinar más a fondo regiones que contengan soluciones atractivas, mientras que las estrategias de diversificación adoptan técnicas que incrementan la capacidad de explorar zonas que no han sido visitadas, buscando soluciones que difieran lo más posible de aquellas que han sido revisadas. Ver [2].

5. ENFOQUE PROPUESTO

A diferencia de los métodos referenciados en el apartado del estado del arte, que trabajan con la técnica de búsqueda tabú para el

modelo básico, utilizando criterios como la tardanza media o tardanza total, en este trabajo se presenta un enfoque de búsqueda tabú para el problema de minimización de la tardanza total ponderada en el modelo básico. La solución propuesta emplea un mecanismo de generación del vecindario mediante intercambios aleatorios pareados.

La secuencia de pasos que sigue el algoritmo de búsqueda tabú propuesto en esta investigación y algunos de los parámetros que se han introducido en él, se pueden resumir como sigue:

- 1. Selección de una solución inicial.** El algoritmo parte de una solución inicial construida aleatoriamente.
- 2. Generación del entorno o vecindario.** Para evolucionar hacia otras soluciones, el algoritmo de búsqueda tabú genera un vecindario de secuencias, que se construye a partir de la solución actual. Esto se consigue eligiendo aleatoriamente un par de trabajos que se intercambian en sus posiciones.
- 3. Evaluación de la función objetivo.** Luego de evaluar cada una de las soluciones del vecindario, se escoge aquella que tenga la menor tardanza total ponderada. Ésta se convierte en la nueva solución actual. Si es necesario, se actualiza el valor de la mejor solución hallada hasta el momento.
- 4. Actualización de la lista tabú.** La actualización consiste en evaluar si la solución de menor tardanza ponderada, hallada anteriormente, se encuentra en la lista tabú, en cuyo caso se remueve ésta del vecindario y se regresa nuevamente al paso anterior. De no estar, la solución actual es añadida a la lista tabú. Mientras no se haya cumplido el criterio de finalización se regresa al paso 2. En el caso aquí desarrollado, el tamaño de la lista tabú es fijo (ver apartado 6).
- 5. Criterio de finalización.** Se ha establecido como criterio de finalización un número máximo de iteraciones, o alternativamente, un número máximo de iteraciones sin disminuir el valor de la mejor solución hallada hasta el momento. Si el criterio de paro es alcanzado, el algoritmo se detiene, de lo contrario, la información es actualizada y se genera una nueva iteración.

6. ANÁLISIS DE RESULTADOS

Para ilustrar la efectividad del algoritmo de búsqueda tabú propuesto en este artículo, se realizaron pruebas con algunos problemas de referencia de 40, 50 y 100 trabajos, tomados de la librería ORLIB (disponible en <http://www.ms.ic.ac.uk/jeb/orlib/wtinfo.html>).

Para cada tamaño se corrieron cinco problemas, tomados de wt100.txt, wt40.txt y wt50.txt, para un total de 15. Los problemas de 100 trabajos aún permanecen sin resolver, razón por la cual en las tablas anexas se encuentra la mejor solución conocida, más no el óptimo, igual que para el problema 19 de 40 trabajos.

A continuación se describen los parámetros utilizados en el algoritmo de búsqueda tabú propuesto:

Vecindario: Como se dijo anteriormente, el vecindario fue construido mediante intercambios aleatorios entre parejas de trabajos. Se hicieron las pruebas con tres tamaños de vecindario diferentes: 60, 132 y 234, los cuales fueron seleccionados de forma que se pudiera comparar el efecto del tamaño del mismo sobre los resultados.

Número de corridas y número de iteraciones: para cada uno de los problemas se hicieron experimentos de 20 corridas, cada una con 5000 iteraciones, para ambos criterios de paro analizados: número máximo de iteraciones y número máximo de iteraciones sin disminuir el valor de la mejor solución hallada hasta el momento.

Tamaño de la lista tabú: A pesar de que no hay nada definitivo en la literatura para establecer este parámetro, se realizaron pruebas con varios tamaños del mismo sin hallarse diferencias significativas en los resultados obtenidos, por tal razón los experimentos fueron realizados con un valor de 5.

Generación de números aleatorios: para poder realizar efectivamente la comparación de los resultados obtenidos con los dos criterios de paro empleados, se utilizaron las mismas secuencias de generación de números aleatorios.

El programa fue implementado en lenguaje Visual Basic 6.0 y las pruebas fueron realizadas en un procesador Intel Pentium 4 de 1.5 GHz.

La tabla 1 muestra los resultados obtenidos para el problema 1 de 40 trabajos en cada tamaño de vecindario y para cada criterio de paro analizado. En la parte inferior se encuentra el óptimo o la mejor solución conocida. Se denota como *Frec Opt* y *%Opt* al número y porcentaje de las veces que se halló el óptimo en las 20 corridas, respectivamente; *t prom corr* al tiempo promedio en segundos que demora el programa en efectuar una corrida y *desv prom opt* a la desviación promedio de los resultados respecto al óptimo o mejor solución encontrada. En el anexo 1 se encuentran los resultados de los 15 problemas analizados.

Para todos los problemas, el criterio de parada *número de iteraciones sin mejorar* siempre resultó mejor o igual que *número de iteraciones*.

A excepción del problema 81 de 50 trabajos, donde la mayor desviación respecto al óptimo es de 12.6%, en todos los demás problemas se obtuvo una desviación menor que el 2.6%, lo cual se considera un resultado bastante aceptable de acuerdo con lo sugerido por diversos autores.

Tabla 1 - Resultados experimentales para el problema 1 de 40 trabajos

Tamaño del Vecindario	Criterio de Paro	Frec Opt	% Opt	t prom corr	desv prom opt	Óptimo
60	Iteraciones	1	5%	15.65	0.019	913
	Iteraciones sin mejorar	1	5%	20.05	0.019	
132	Iteraciones	4	20%	39.6	0.016	
	Iteraciones sin mejorar	4	20%	53.4	0.016	
234	Iteraciones	3	15%	87.7	0.02	
	Iteraciones sin mejorar	3	15%	111.45	0.02	

En los resultados no se presenta ninguna tendencia clara entre el número de veces que se encuentra la solución óptima y el tamaño del vecindario. Se observa este mismo comportamiento para la desviación respecto al óptimo. A partir de esto se hizo una prueba piloto con el problema 65 de 50 trabajos, aumentando el número de corridas hasta 200 para obtener un mayor número de observaciones. En la tabla 2 se observa que a medida que aumenta el número de corridas, el vecindario de tamaño 132 muestra un mejor comportamiento. También es importante notar que el vecindario de tamaño 234 presenta el peor comportamiento de todos los vecindarios analizados. Lo anterior muestra que la utilización de este vecindario no se justifica, debido a que éste toma un tiempo de ejecución más largo.

En la figura 1 se presenta un histograma que clasifica el número de iteraciones consecutivas sin mejorar para las corridas de todos los problemas de prueba utilizados. Este número sólo considera el mayor valor alcanzado antes de cumplirse el límite de 5000 iteraciones sin mejorar.

Tabla 2. Número de soluciones óptimas encontradas variando el número de corridas

Número de corridas	Tamaño del vecindario		
	60	132	234
20	1	1	0
50	1	3	0
75	2	4	0
100	3	4	0
125	4	4	0
150	4	4	0
200	5	6	2

El 38% de las veces dicho valor ocurre en una iteración menor o igual a la número 337. De hecho, el 82% de las veces este valor no llega a ser mayor de 2330. Lo anterior deja ver que sería suficiente un número de corridas para el criterio no mayor a 2330.

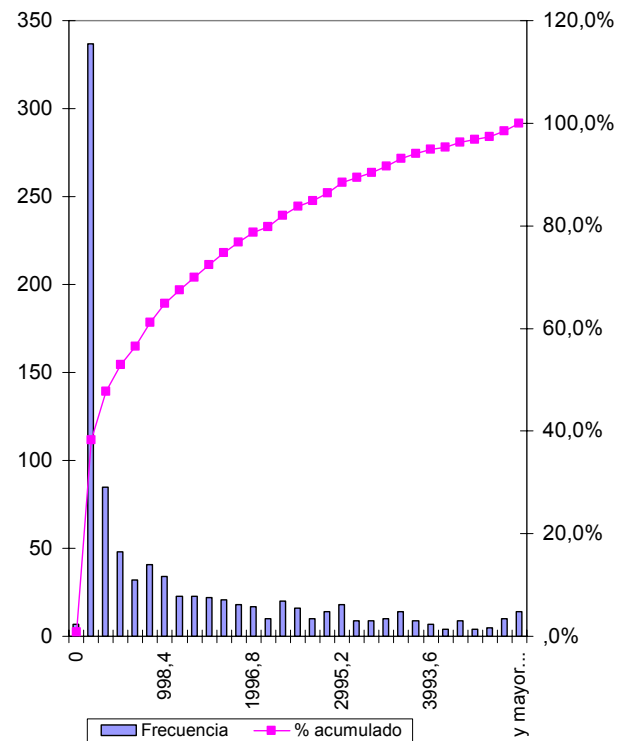


Figura 1. Histograma del número de iteraciones consecutivas sin mejorar

7. CONCLUSIONES

A diferencia de otros trabajos, en la investigación aquí desarrollada se presenta la búsqueda tabú como una alternativa para solucionar el problema del modelo básico, cuando se trabaja con el criterio de minimizar la tardanza total ponderada. Este problema ha sido tratado por muchos autores, empleando para ello técnicas como la búsqueda local, colonia de hormigas, algoritmos genéticos, métodos constructivos y enfriamiento simulado; pero aplicado a otros criterios de ejecución.

Respecto al criterio del número de iteraciones sin mejorar, los resultados del estudio sugieren que se puede disminuir el valor de este parámetro sin afectar la calidad de las soluciones encontradas.

Los resultados muestran que el algoritmo produce soluciones óptimas o cercanas al óptimo en todas las instancias probadas.

A pesar de que los experimentos con 20 corridas no muestran diferencia significativa en las frecuencias de hallazgos del óptimo, la prueba piloto muestra que el vecindario de 234 presenta los peores resultados y el vecindario de 132 presenta un comportamiento levemente superior que el de 60.

8. TRABAJO FUTURO

De acuerdo a los resultados obtenidos al realizar las pruebas, se propone como parte de una futura investigación por parte del grupo, emplear técnicas enfocadas a la diversificación con el ánimo de explorar regiones no analizadas. Esto podría mejorar la calidad de las soluciones encontradas. Así mismo, a partir de la estimación de un número de iteraciones sin mejorar que sea suficiente para llegar a la solución óptima, es de interés evaluar la sensibilidad de este parámetro en la eficiencia del algoritmo.

Con el fin de escapar de los óptimos locales, se propone implementar movimientos y secuencias aleatorias cuando se alcance un valor crítico de iteraciones sin mejorar.

Buscando mejorar el desempeño computacional, se implementará el algoritmo en lenguaje C.

Por último, el grupo planea implementar técnicas adicionales como algoritmos genéticos, colonia de hormigas, tanto para el problema del modelo básico, como para el problema de programación de job shop.

9. AGRADECIMIENTOS

El grupo agradece a la Dirección de Investigación Medellín (DIME), de la Universidad Nacional de Colombia, por el apoyo brindado para el desarrollo de la presente investigación.

10. REFERENCIAS

- [1] T.S. Abdul-Razaq , C. N. Potts, L. N. Van Wassenhove. “*A survey of algorithms for the single machine total weighted tardiness scheduling problem*”. **Discrete Applied Mathematics**. (1990).
- [2] R. B. Acero, J.F Torres. “*Aplicación de una heurística de búsqueda tabú en un problema de programación de tareas en línea flexible de manufactura*”. Universidad de los Andes. [en línea]
<<http://empleados.uniandes.edu.co/dependencias/Departamentos/ingenieriaindustrial/magister/memos/sep2002/ACERAHB.pdf>> (Consulta: 14 de noviembre de 2003)
- [3] M. Besten, T. Stützle, M. Dorigo. “*Design of Iterated Local Search Algorithms: An example application to the single machine total weighted tardiness problem*”. **Proceedings of EvoWorkshops, LNCS**. Springer, Berlin. (2001).
- [4] M. Besten, T. Stützle, M. Dorigo. “*Ant Colony Optimization for the Total Weighted Tardiness Problem*”. **Parallel Problem Solving from Nature: 6th international conference**, volume 1917 of LNCS, Berlin, p. 611-620. (2000).
- [5] B. Bilgen, I. Ozkarahan. “*Tabu Search Technique for Single Machine Scheduling Problem*”. **Proc. of Decision Sciences Institute's National Conf.**, p. 1104. (2000).
- [6] H. A. Crauwels, C. N. Potts, L. N. Van Wassenhove. “*Local search heuristics for the single machine total weighted tardiness scheduling problem*”. **INFORMS Journal on Computing**. (1998).
- [7] F. Glover and M. Laguna. “*Tabu Search, in Modern Heuristic Techniques for Combinatorial Problems*”. C.R. Reeves, editor, John Wiley & Sons, Inc. (1993).
- [8] A. Islam, M. Eksioglu (1997). “*A tabu search approach for the single machine mean tardiness problem*”. **Journal of the Operational Research Society**, 48, p. 751–755.
- [9] A M. Laguna, A J. W. Barnes, A F. Glover. “*Tabu Search Methodology for a Single Machine Scheduling Problem*”. **J. of Int. Manufacturing**, 2, p. 63-74. (1991).
- [10] H. Matsuo, C. J. Suh, R. S. Sullivan. “*A controlled search simulated annealing method for the single machine weighted tardiness problem*”. Working paper. Department of Management, University of Texas. (1987).
- [11] M. Pinedo. “*Scheduling Theory Algorithms and Systems*”. Prentice Hall, p. 44-46. (1995).
- [12] C. N. Potts, L. N. Van Wassenhove. “*Single Machine tardiness sequencing heuristics*”. **IIE transactions**. (1991).
- [13] C.N. Potts, and L.N. Van Wassenhove. “*A branch and bound algorithm for total weighted tardiness problem*”. **Operations Research**, 33, p. 363–377. (1985).
- [14] R.H. Storer, S. Avci, S. Akturk,. “*A problem space algorithm for the single machine weighted tardiness problems*”. **IIE Transactions** 35, p. 479-486. (2003).
- [15] R.H. Storer, S.D. Wu, and R. Vaccari. “*Local search in problem and heuristic space for job shop scheduling*”. **ORSA Journal on Computing**, 7, p. 453–467. (1995).
- [16] R.H. Storer, S.D. Wu, and R. Vaccari. “*New search spaces for sequencing problems with application to job shop scheduling*”. **Management Science**, 38, p. 1495–1509. (1992).

ANEXO 1. Resultados Problemas de Prueba

Vecindario: Variable
 Número de corridas: 20
 Número de iteraciones: 5000

Frec Opt: número de veces que halló el óptimo
 % Opt: porcentaje de las veces que halló el óptimo
 t prom corr: tiempo promedio por corrida
 desv prom ópt: desviación respecto al óptimo

Cantidad Trabajos	Problema	Tamaño Vecindario	Criterio de Paro	Frec Opt	% Opt	t prom corr	desv prom opt	Optimo
40	1	60	I	1	5%	15.65	0.019	913
			ISM	1	5%	20.05	0.019	
		132	I	4	20%	39.6	0.016	
			ISM	4	20%	53.4	0.016	
		234	I	3	15%	87.7	0.02	
			ISM	3	15%	111.45	0.02	
	19	60	I	2	10%	21.9	0.0036	77122*
			ISM	2	10%	23.4	0.0036	
		132	I	7	35%	54.23	0.0026	
			ISM	7	35%	58.9	0.0026	
		234	I	2	10%	119.5	0.0034	
			ISM	2	10%	123.45	0.004	
	25	60	I	6	30%	19.7	2,06E-02	128747
			ISM	10	50%	31.65	1,00E-02	
		132	I	20	100%	49.1	0	
			ISM	20	100%	50.8	0	
		234	I	20	100%	109.1	0	
			ISM	20	100%	110.79	0	
	100	60	I	2	10%	23.45	3,34E-02	157296
			ISM	4	20%	39.15	2,42E-02	
		132	I	16	80%	59.06	4,45E-5	
			ISM	16	80%	62.1	4,45E-5	
		234	I	18	90%	126.3	2,2E-5	
			ISM	18	90%	128.85	2,2E-5	
125	60	I	0	0%	15.74	0.0001	104531	
		ISM	4	20%	30.1	8,33E-02		
	132	I	20	100%	40.21	0		
		ISM	20	100%	42.7	0		
	234	I	20	100%	89.2	0		
		ISM	20	100%	91.2	0		

Cantidad Trabajos	Problema	Tamaño Vecindario	Criterio de Paro	Frec Opt	% Opt	t prom corr	desv prom opt	Optimo
50	11	60	I	18	90	19.1	0.0009	43504
			ISM	18	90	21.9	0.0009	
		132	I	15	75	46.9	0.002	
			ISM	15	75	51.5	0.002	
		234	I	13	65	96.7	0.003	
			ISM	13	65	106.2	0.003	
	47	60	I	0	0%	24.5	0.00012	133289
			ISM	0	0%	45.1	0.00010	
		132	I	18	90%	60.2	4,35E-5	
			ISM	18	90%	76.35	4,35E-5	
		234	I	16	80%	123.1	8,7E-5	
			ISM	16	80%	134.2	8,7E-5	
	56	60	I	20	100%	24.5	0	1258
			ISM	20	100%	25.3	0	
		132	I	20	100%	59.3	0	
			ISM	20	100%	65.3	0	
		234	I	20	100%	122.1	0	
			ISM	20	100%	135.15	0	
	81	60	I	17	85%	19.6	0,076	816
			ISM	17	85%	25.9	0,076	
		132	I	15	75%	47.54	0,126	
			ISM	15	75%	60.95	0,126	
		234	I	19	95%	97.2	0,025	
			ISM	19	95%	132.2	0,025	
92	60	I	0	0%	23.1	0,0009	66340	
		ISM	0	0%	41.6	0,0007		
	132	I	14	70%	57.8	0,001		
		ISM	14	70%	74.75	0,001		
	234	I	8	40%	123.02	0,002		
		ISM	8	40%	126.35	0,002		

Cantidad Trabajos	Problema	Tamaño Vecindario	Criterio de Paro	Frec Opt	% Opt	t prom corr	desv prom opt	Optimo
100	4	60	I	10	50%	33.2	0.011	5011*
			ISM	10	50%	39.05	0.011	
		132	I	14	70%	78.15	0.005	
			ISM	14	70%	89.85	0.005	
		234	I	10	50%	150.01	0.012	
			ISM	10	50%	156.55	0.011	
	10	60	I	0	0%	46.5	0,007	53208*
			ISM	0	0%	59.7	0,006	
		132	I	0	0%	107.2	0,0053	
			ISM	0	0%	147	0,0052	
		234	I	0	0%	206	0,007	
			ISM	0	0%	299.7	0,006	
	89	60	I	0	0%	43.5	0,025	54612*
			ISM	1	5%	83.95	0,023	
		132	I	0	0%	101.4	0,023	
			ISM	0	0%	246	0,021	
		234	I	0	0%	192.9	0,026	
			ISM	0	0%	338.6	0,024	
	90	60	I	0	0%	94.8	0,02	75061*
			ISM	0	0%	40.05	0,018	
		132	I	0	0%	100.45	0,002	
			ISM	0	0%	221.05	0,001	
		234	I	0	0%	202.3	0,001	
			ISM	2	10%	516.35	0,001	
115	60	I	0	0%	46.7	0.019	70190*	
		ISM	0	0%	98.75	0.016		
	132	I	0	0%	105.6	0.003		
		ISM	0	0%	238.5	0.002		
	234	I	8	40%	203.5	0.0005		
		ISM	14	70%	382.1	0.0004		